OS 444 GROUP 36

MAY 6, 2018

PROJECT 2: I/O ELEVATORS

GROUP 36

SCOTT RUSSELL
ARYA ASGARI
FISCHER JEMISON

CONTENTS

1 Design Plan 2

2 Version Control: Table 2

3 Work Log: What was done when? 3

4 Assignment Questions 3

4.1 Main Point of Assignment . . . 3

4.2 Approach to Problem . . . 3

4.3 Ensuring Correctness . . . 3

4.4 What we learned . . . 3

4.5 How TA should evaluate work . . . 4

1 Design Plan

Our designed stemmed from an open source baseline implementation of the no-op scheduler found here: "https://elixir.bootlin.com/l
iosched.c" If you compare that base implementation to ours you can see that very little was changed in the grand scheme of things. Mainly the creation of the Forward/Back head and being able to step through the queue up and down the elevator. We decided to go with the LOOK implementation instead of C-Look for simplicity. Rather than having to only drop off in a single direction and 'drop' down to reset after each elevator lift having the ability to go in both directions made creation the forward/backwards directions in the queue simpler as it was reversible.

2 Version Control: Table

Commit Hash	**Commit Description**	**Date Added**
bef7896	Initial sstf-iosched.c file	Fri May 4th 2018 -0700
182072c	readme for building sstf	Fri May 4th 2018 -0700
adaa8b2	update readme	Fri May 4th 2018 -0700
8f4f083	delete readme.md (old one)	Fri May 4th 2018 -0700
8273541	Added documentation for io schedulers	Sat May 5th 2018 -0700
80c6ffa	Adds test stuff and tweaks README	Sat May 5th 2018 -0700
70c5fce	removes test files	Sat May 5th 2018 -0700
9bd63b3	update sstf-iosched.c	Sun May 6th 2018 -0700
0b86c99	Updates README	Sun May 6th 2018 -0700
75c1040	Adds block.patch file	Sun May 6th 2018 -0700

3 Work Log: What was done when?

May 2nd	Work Done
Scott Russell	Researched sstf scheduler and comparison to no-op.
Arya Asgari	Researched no-op scheduler and built-in elevator
Fischer Jemison	Additional research into sstf scheduler as well as kernel building debugging (problems with file access permissions)

May 5th	Work Done
Scott Russell	Worked on implementation of sstf scheduler with ReadMe.
Arya Asgari	Created patch file of sstf file against no-op scheduler, worked on latex documentation.
Fischer Jemison	Created test file to show functionality of code with many ptread calls.

4 Assignment Questions

4.1 Main Point of Assignment

It seems like the primary objective of the assignment is to start developing the skills necessary to work with the kernel on a low level. Also, this assignment helps build a better understanding of elevator algorithms and the Linux I/O scheduler. Much of this assignment required researching and understanding the no-op I/O scheduler and how the built in elevator already worked.

4.2 Approach to Problem

Allot of the project was thinking about how to implement the scheduler. Looking between the No-Op elevator that I adjusted there was only about 40-50 lines of code changed. This is an example of a time not task assignment. Conceptually thinking about the problem turned into the longest section. We spent our first entire meeting simply thinking about how we were going to implement the sstf file. A small portion was adjusting files and creating the patch file.

4.3 Ensuring Correctness

To test the functionality of our scheduler we make a large number of threads that write one thousand character strings to bog down the I/O functionality and to be able to test our sstf scheduler.

4.4 What we learned

We learned a great deal while doing this assignment. Since much of this project required research, we spent a fair bit of time understanding the no-op I/O scheduler on a deeper level. We also wanted to make sure we understood the LOOK

elevator algorithm, as well as the built in elevator framework. For checking our implementation, we learned how to create a patch file and apply that to the no-op scheduler.

4.5 How TA should evaluate work

To evaluate our work, please see the README in our submission.